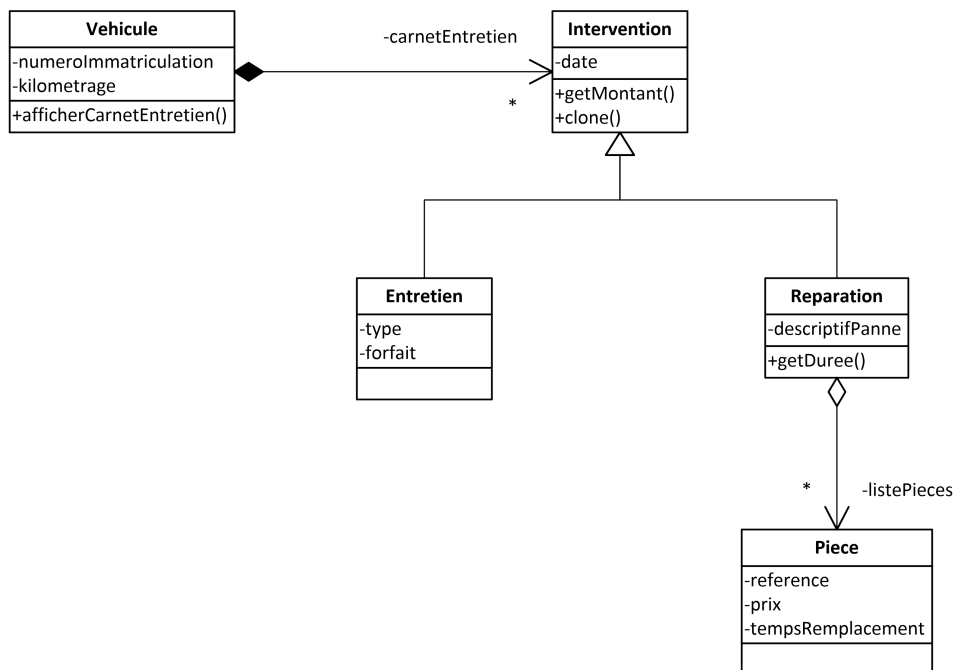


## CPOOA

### TP noté 2016-2017

3h, documents de cours et de TP/TD autorisés

Le but de ce TP consiste à implémenter le diagramme de classes ci-dessous qui modélise une partie d'un système de gestion des interventions pratiquées dans un garage. Toutes les classes seront développées selon le modèle standard (constructeurs, getters, setters, opérateurs, etc.), sauf indications contraires. Le diagramme ne montre pas les méthodes standard, ni les méthodes qui doivent être redéfinies.



L'usage de la STL est obligatoire pour gérer les collections. Le « main » donné en fin de sujet doit servir de base à votre travail et servira à la notation. Il vous est donné dans le fichier « Princ.cpp » ainsi que la classe Date dans le fichier « Date.h ».

#### Exercice 1 : Piece (4 points)

Une pièce (au sens pièce mécanique du terme) est caractérisée par une référence (**string**), un prix (**double**) et une durée de remplacement en minutes (**int**). La référence est un texte libre (non vide), non modifiable. Si une référence vide est indiquée, une exception doit être levée. Le prix doit être naturellement strictement positif. Dans le cas contraire une exception doit être levée. La durée doit être comprise entre 15 et 360 minutes. Dans le cas contraire une exception doit être levée. Deux pièces sont égales si tous les champs sont égaux. Ces trois informations sont toujours nécessaires pour construire un objet de cette classe.

Implémentez cette classe en respectant le standard et en lançant des **InterventionException** si les paramètres des méthodes ne respectent pas les contraintes ci-dessus. Cette classe d'exception est à écrire par vous-même. Dans les classes suivantes le contrôle de validité par exceptions n'est pas demandé afin de simplifier le sujet.

#### Exercice 2 : Intervention et Entretien (6 points)

Une intervention est une notion qui couvre les différentes activités qu'un garage peut avoir sur un véhicule. On distingue, pour l'instant, les entretiens courants (révisions,

plaquettes, vidanges, etc.) et les réparations (les pannes, les accidents, etc.). Une intervention est uniquement caractérisée par une date (classe **Date** fournie). On peut également consulter le montant d'une intervention (**getMontant()**), même si dans cette classe on ne sait pas encore le calculer. De même, on veut pouvoir dupliquer une intervention (**clone()**).

Un entretien est une intervention relativement simple. Un entretien est caractérisé par un type (**int**) et un forfait (**double**). Dans ce cas, le montant de l'intervention est tout simplement le forfait. Pour le type vous devrez utiliser des constantes ou des valeurs d'enum.

### Exercice 3 : Réparation (7 points)

Une réparation consiste, pour simplifier, à remplacer une ou plusieurs pièces. La durée de la réparation (méthode **getDuree()**), en minutes, est la somme des durées de remplacement des pièces. Le montant d'une réparation est calculé en faisant la somme du prix des pièces, et en y ajoutant le prix de la main d'œuvre. Une heure (soit 60 minutes...) de main d'œuvre coûte 150€... Une heure commencée est à payer entièrement.

Attention, quand on ajoute une pièce à une réparation, on ne duplique pas la pièce. De même, quand on duplique une réparation, on ne duplique pas les pièces !

### Exercice 4 : Véhicule (3 points)

Un véhicule est caractérisé par son numéro d'immatriculation (**string**) et son kilométrage (**int**). Si le numéro d'immatriculation est toujours nécessaire pour enregistrer un véhicule, le kilométrage n'est pas toujours connu. On doit pouvoir ajouter une intervention à un véhicule et la supprimer. La fonction **afficherCarnetEntretien()** affiche toutes les interventions ainsi que le montant total des interventions sur le véhicule.

Le main suivant doit s'exécuter :

```
#include <iostream>

#include "Piece.h"
#include "Date.h"
#include "Reparation.h"
#include "Entretien.h"
#include "Vehicule.h"

int main() {
    Piece * p1 = new Piece("Courroie transmission", 130., 25);
    std::cout << "p1 = " << *p1 << std::endl;

    Piece * p2 = new Piece("Alternateur", 245., 45);
    std::cout << "p2 = " << *p2 << std::endl;

    Piece * p3;

    try {
        Piece * p3 = new Piece("Boîte de vitesses", 245., 0);
    }
    catch (InterventionException e) {
        std::cout << e.what () << std::endl;
    }

    Date d1(23, Date::octobre, 2014);
    std::cout << "d1 = " << d1 << std::endl;

    Reparation * r1 = new Reparation(d1, "Pb d'alternateur A4");
    std::cout << "r1 = " << *r1 << std::endl;
    r1->add(p1);
    r1->add(p2);
    std::cout << "r1 = " << *r1 << std::endl;

    Intervention * r2 = r1->clone();
    std::cout << "r2 = " << *r2 << std::endl;

    Entretien *e1 = new Entretien( Date(11, Date::mars, 2015),
                                   Entretien::Vidange, 450.);
    std::cout << "e1 = " << *e1 << std::endl;

    Intervention *e2 = e1->clone();
    std::cout << "e2 = " << *e2 << std::endl;

    Vehicule *A4 = new Vehicule ("FL309DE", 8234);
    std::cout << "A4 = " << *A4 << std::endl;

    A4->add(e1);
    A4->add(r1);

    A4->afficherCarnetEntretien(std::cout);

    system("pause");
}
```

et donner ce résultat :

```
\\vmware-host\Shared Folders\Documents\Enseignement\Années\2016-2017\L3\CPOOA\TP Noté\Atelier\Deb
p1 = Piece [reference=Courroie transmission, prix=130.000000, tempsRemplacement=25]
p2 = Piece [reference=Alternateur, prix=245.000000, tempsRemplacement=45]
Temps de remplacement incorrect
d1 = 23/10/2014
r1 = Reparation [date=23/10/2014;descriptif=Pb d'alternateur A4;montant=0.000000;duree=0]
r1 = Reparation [date=23/10/2014;descriptif=Pb d'alternateur A4;montant=675.000000;duree=70]
r2 = Reparation [date=23/10/2014;descriptif=Pb d'alternateur A4;montant=675.000000;duree=70]
e1 = Entretien [date=11/3/2015, type = 71, forfait = 450.000000]
e2 = Entretien [date=11/3/2015, type = 71, forfait = 450.000000]
A4 = Vehicule [immatriculation=FL309DE;kilometrage=8234]
Carnet d'entretien
-----
Entretien [date=11/3/2015, type = 71, forfait = 450.000000]
Reparation [date=23/10/2014;descriptif=Pb d'alternateur A4;montant=675.000000;duree=70]
-----
Cout total : 1125
-----
```